1.Use a HashSet to hold Employee Objects. Upon running the application, the details of the employees added to the HashSet should be displayed.

Employee <>

|-- id

|-- name

|-- salary

|-- department

|-- displayDetails()

Feel free to add properties and methods to Employee Class Note: if we try to store any object other than Employee Object in HashSet, we should not be allowed to.

**import** java.util.HashSet;

**import** java.util.Set;

**import** java.util.\*;

**class** Employees

{

**int** empid,empSalary;

String empName,empDept;

**public** Employees(**int** empid,String empName, **int** empSalary, String empDept)

{

**super**();

**this**.empid=empid;

**this**.empName=empName;

**this**.empSalary=empSalary;

**this**.empDept=empDept;

}

@Override

**public** String toString() {

**return** "Employees [empid=" + empid + ", empSalary=" + empSalary + ", empName=" + empName + ", empDept=" + empDept

+ "]";

}

}

**public** **class** Hashset {

**public** **static** **void** main(String[] args)

{

Employees e1=**new** Employees(221,"Aparna",50000,"Consultancy");

Employees e2=**new** Employees(222,"Deepika",70000,"IT");

Employees e3=**new** Employees(223,"Havi",65000,"Banking");

Employees e4=**new** Employees(223,"Moni",75000,"Finance");

Set<Employees> employees= **new** HashSet<>();

employees.add(e1);

employees.add(e2);

employees.add(e3);

employees.add(e4);

**for** (Employees employees2 : employees) {

System.***out***.println(employees2);

}

}

}

2. Write an application to hold 10 random int values as keys and 10 random double values as values for a HashMap. Print the data store in the HashMap. Note: Keys can only be int and values double

**import** java.util.\*;

**public** **class** Hashmap {

**public** **static** **void** main(String[] args)

{

HashMap<Integer,Double> map=**new** HashMap<Integer,Double>();

map.put(10, 10.00);

map.put(20, 20.00);

map.put(30, 30.00);

map.put(40, 40.00);

map.put(50, 50.00);

map.put(60, 60.00);

map.put(70, 70.00);

map.put(80, 80.00);

map.put(90, 90.00);

map.put(100, 100.00);

**for**(Map.Entry<Integer, Double> e: map.entrySet())

System.***out***.println("Key sn: "+e.getKey()+" Value : "+e.getValue());

}

}

3.Write a generic method to exchange the positions of two different elements in an array.

**class** Generic<T>{

**public** T[] swap(T[] a, **int** i, **int** j) {

T temp = a[i];

a[i] = a[j];

a[j] = temp;

**return** a;

}

}

**public** **class** Generic3 {

**public** **static** **void** main(String[] args) {

Integer arr[] = {1,2,3,4,5};

Generic<Integer> objGeneric = **new** Generic<>();

Integer[] arr1=objGeneric.swap(arr, 0, 3);

**for**(**int** i : arr1) {

System.***out***.println(i);

}

}

}

4.

**class** Generic<K, V>

{

K Key;

V value;

**public** K getKey() {

**return** Key;

}

**public** **void** setKey(K key) {

**this**.Key = key;

}

**public** V getValue() {

**return** value;

}

**public** **void** setValue(V value) {

**this**.value = value;

}

}

**public** **class** Generic4 {

**public** **static** **void** main(String[] args)

{

Generic<String, String> ob1=**new** Generic<>();

ob1.setKey("1");

ob1.setValue("Hello");

Generic<String, java.util.Date> ob2=**new** Generic<>();

ob2.setKey("Today is");

ob2.setValue(**new** java.util.Date());

System.***out***.println(ob1.getKey()+" "+ob1.getValue());

System.***out***.println(ob2.getKey()+" "+ob2.getValue());

}

}